CHEAP BEEPS - EFFICIENT SYNTHESIS OF SINUSOIDS AND SWEEPS IN THE MDCT DOMAIN

Sascha Disch[1], Benjamin Schubert[1], Bernd Edler[2]

[1]Fraunhofer Institute for Integrated Circuits IIS, Erlangen, Germany
[2]International Audio Laboratories Erlangen, Erlangen, Germany

ABSTRACT

Modern transform audio coders often employ parametric enhancements, like noise substitution or bandwidth extension. In addition to these well-known parametric tools, it might also be desirable to synthesize parametric sinusoidal tones in the decoder. Low computational complexity is an important criterion in codec development and essential for acceptance and deployment. Therefore, efficient ways of generating these tones are needed. Since contemporary codecs like AAC or USAC are based on an MDCT domain representation of audio, we propose to generate synthetic tones by patching tone patterns into the MDCT spectrum at the decoder. We demonstrate how appropriate spectral patterns can be derived and adapted to their target location in (and between) the MDCT time/frequency (t/f) grid to seamlessly synthesize high quality sinusoidal tones including sweeps.

Index Terms— Sinusoid, Sweep, Codecs, Parametric Audio Coding, Signal Synthesis

1. INTRODUCTION

In low bit rate transform audio coders, it might be desirable to additionally synthesize sinusoidal tones in the decoder from compact parametric side information [1]. Straightforward implemented, a simple bank of oscillators runs in parallel with the decoder followed by a mixing in time domain. However, such oscillators at high sampling rate are a huge computational burden. Computational complexity is an important criterion in codec development and deployment, therefore more efficient ways of generating these tones are needed.

Contemporary codecs like Advanced Audio Coding (AAC) [2] or Unified Speech and Audio Coding (USAC) [3] are based on a Modified Discrete Cosine Transform (MDCT). Consequently, we propose to generate synthetic tones by directly patching tone patterns into the MDCT spectrum at the decoder. Previous publications in the field relate to synthesis of sinusoidal tones directly in time domain, or piecewise constant tones in DFT frequency domain [4], and to the SNR optimization of truncated patterns in the DFT domain [5]. The embedding of piecewise constant frequency tones based on MDCT spectra in a perceptual codec environment [6] or a bandwidth extension scenario [7] has already been described. However, the efficient generation of sweeps and their linkage to seamless tracks in MDCT domain has seemingly not been adressed yet, nor has the definition of sensible restrictions on the available degrees of freedom in the parameter space. Most important, only these restrictions essentially enable an ultra-low complexity implementation as proposed in this paper.

2. MDCT BASICS

The MDCT of a real signal $x(n)$ is defined for signal segments of length N, windowed with $w(n)$ at time l, with $M = \frac{N}{2}, m \in \{0, 1, \ldots, M-1\}$,

$$\begin{aligned} & X_{MDCT}(l,m) \\ & = MDCT\{w_a(n) \cdot x(l,n)\} \\ & = \sqrt{\tfrac{2}{M}} \sum_{n=0}^{N-1} w_a(n) \cdot x(l,n) \cos(\tfrac{\pi}{M}(m+\tfrac{1}{2})(n+\tfrac{1}{2}+\tfrac{M}{2})) \end{aligned} \quad (1)$$

its inverse, with $N = 2M, n \in \{0, 1, \ldots, N-1\}$, by

$$\begin{aligned} & \tilde{x}(l,n) \\ & = IMDCT\{X(l,m)\} \\ & = w_s(n) \sqrt{\tfrac{2}{M}} \sum_{m=0}^{M-1} X(l,m) \cos(\tfrac{\pi}{M}(m+\tfrac{1}{2})(n+\tfrac{1}{2}+\tfrac{M}{2})) \end{aligned} \quad (2)$$

The MDCT can be seen as the real part of the Complex Modified Discrete Cosine Transform (CMDCT). Moreover, the CMDCT can be expressed as an Oddly-Stacked Discrete Fourier Transform (ODFT) or Discrete Fourier Transform (DFT) and exponential pre- and post-twiddling phase terms

$$\begin{aligned} & X_{CMDCT}(l,m) \\ & = CMDCT\{w_a(n) \cdot x(l,n)\} \\ & = ODFT\{w_a(n) \cdot x(l,n)\} \cdot e^{-j\frac{\pi}{M}(m+\frac{1}{2})(\frac{1}{2}+\frac{M}{2})} \\ & = DFT\{w_a(n) \cdot x(l,n) \cdot e^{-j\frac{\pi}{2M}n}\} \cdot e^{-j\frac{\pi}{M}(m+\frac{1}{2})(\frac{1}{2}+\frac{M}{2})} \end{aligned} \quad (3)$$

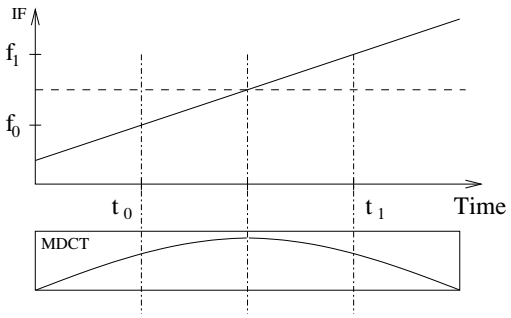

Fig. 1. Parameter alignment of sinusoidal pattern wrt. MDCT time block.

3. TONE PATTERNS

3.1. MDCT Peculiarities

As can be seen from Equations 3, which contain an exponential so-called post-twiddle term, the basis functions of the CMDCT have a time shift in comparison to DFT or ODFT. If a decoupling of the absolute phase offset φ_0 of the patched sinusoids from the actual spectral position of patch application is required, this twiddle must be taken into account. We propose to do the pattern extraction and the patching in the ODFT domain and post-process the superposition of all patterns by application of said twiddle before the mixing with the MDCT coefficients. Each patch is obtained by extracting truncated complex ODFT spectra of prototypical sinusoids or sweeps generated according to the following equations. A sinusoid with varying instantaneous frequency (IF) $f(t)$ can be synthesized as

$$x(t) = cos\left(\varphi(t)\right) \tag{4}$$

with the instantaneous phase

$$\varphi(t) = \varphi(0) + \int_0^t 2\pi f(\tau) d\tau \tag{5}$$

For simplicity of the relation between time discrete MDCT and time continuous sinusoid description, a normalized sampling rate $f_s = 1$ is assumed in the following. The instantaneous frequency (IF) $f(\tau)$ of the sweep templates is chosen such that start and target IF are exactly reached at the time domain aliasing cancellation (TDAC) symmetry points $t_0 = N/4 + 0.5$ and $t_1 = 3N/4 + 0.5$ of each MDCT time block of length N (see Figure 1). A linear sweep from frequency f_0 to f_1 spanning a frequency range $\Delta f = f_1 - f_0$ in a time interval of length $M = N/2$ has an IF

$$f(t_0 + t) = f_0 + \frac{\Delta f}{M} t \tag{6}$$

leading to an instantaneous phase

$$\varphi(t_0 + t) = \varphi(t_0) + 2\pi f_0 t + \frac{\pi \Delta f}{M} t^2 \tag{7}$$

rotation	operation	implementation
0	1	copy "0" pattern
$\frac{\pi}{2}$	i	swap $\Re$ and $-\Im$ part of "0" pattern
π	-1	negate "0" pattern
$\frac{3\pi}{2}$	$-i$	swap $-\Re$ and $\Im$ part of "0" pattern
$\frac{\pi}{4} + \frac{n\pi}{2}$	dto.	do the above on "$\frac{\pi}{4}$" pattern

Table 1. Operations for simple rotations.

Sinusoids with start and end frequencies of doubled resolution (compared to the MDCT to be employed for pattern synthesis) can be generated by selecting $f_0 = k/(4M)$ and $f_1 = (k + m)/(4M)$, with frequency offset m measured in transform bin indices. Odd indices correspond to "on-bin" frequencies and even indices give "between-bin" frequencies. The phase progress between subsequent frames can be computed as

$$\Delta\varphi = \varphi(t_1) - \varphi(t_0) = 2\pi f_0 M + \pi \Delta f M = k\frac{\pi}{2} + m\frac{\pi}{4} \tag{8}$$

This means that for seamless temporal chaining of patterns, the phase of each patch has to be adjusted by an integer multiple of $\frac{\pi}{4}$ depending on the start frequency index k and the frequency offset index m of the preceding pattern. The variable m can also be seen as the sweep rate, where e.g. $m = 1$ denotes a half-bin sweep over the duration of one time block.

The initial spectral position of these prototypical sinusoid or sweep patterns should be set to $M/2$ in order to minimize cyclic folding errors. Dependent on the spectral distance d of prototypical sinusoid and patching target location, the patch is adapted by post-processing rotations of $d\pi/2$ to always obtain a predefined fixed phase independent of patching target location. In other words, a post-processing rotation compensates for the unwanted phase rotation that is inherently caused by the spectral shift.

3.2. Efficiency and Accuracy Considerations

To keep the amount of stored patterns reasonably small and, most important, to be able to exploit the fact that rotations by certain simple fractions of π can be attained by the trivial operations listed in Table 1, it is necessary to restrict the possible frequencies and sweeps. These restrictions should be applied such that still a perceptually satisfactory reproduction of the parametrically coded signal parts is possible. Since such a signal part can consist of an arbitrary time sequence of tone patterns, each additional degree of freedom multiplies the number of patterns to be stored or, alternatively, the computational costs for adaptation of the patterns. Thus it makes good sense to choose the spectral resolution such that no detuning effect is perceived by the average listener in the intended target spectral range. It is well known that trained lis-

teners and musicians are able to perceive detunings down to 5 cents, while the average listener might accept deviations of approx. 10 cents, a tenth of a semi-tone [8]. Therefore, the spectral replacement of sine tones should only be done above a certain cut-off frequency that corresponds to the worst-case scenario of allowable detuning. For example in a 512 band MDCT, at a sampling frequency of 12.8 kHz, the spectral resolution per band is 12.5 Hz. Choosing half-band resolution for the tone patterns, the maximum frequency deviation amounts to 3.125 Hz, which is equal or below 10 cents above a cut-off frequency of approx. 540 Hz.

The patterns to be stored are truncated. The actual size of the patterns depends on the window type that is usually already determined by the transform coder (e.g. sine or KBD window for AAC) and the allowable signal-to-noise ratio (SNR). Although complex valued patterns are stored, the actual patching is only done using the real part of the fittingly rotated pattern.

3.3. Tone Patterns

For the aforementioned reasons the spectral resolution was chosen twice the nominal resolution of the MDCT. As a consequence, two versions of all pattern need to be stored, one for sinusoids with frequencies that coincide with a bin position (on-bin pattern) and one for frequencies that are located between bin positions (between-bin pattern). For smallest possible memory requirements, the patterns symmetry can be exploited by storing only half of the coefficients of the actual pattern.

According to Equation 8 (setting $m = 0$), in any time sequence consisting of stationary sine tone patterns only, the wrapped phase progress amounts to $\Delta\varphi = \pi/2$ or $\Delta\varphi = -\pi/2$ for on-bin patterns, and $\Delta\varphi = 0$ or $\Delta\varphi = \pi$ for between-bin patterns. This is due to the odd frequency stacking of the MDCT.

The absolute wrapped phase can be calculated by $\varphi_0 + n\pi/2$ with n as an integer number $\in \{1, 3\}$ for on-bin patterns and $\in \{2, 4\}$ for between-bin patterns. The choice of the actual integer number depends on the parity of the bin number (even/odd). φ_0 denotes an arbitrary phase offset value. Hence, for purely stationary tone patterns, a post-processing by four alternative rotations is needed in order to fit the patterns to their intended position in the t/f grid of a sequence of MDCT spectra. A choice of $\varphi_0 = n\pi/2, n \in \mathbb{N}$ renders these rotations trivial.

Also, two versions of each sweep pattern needs to be stored, one for sweeps with start frequencies that coincide with a bin position and one for start frequencies that are located between bin positions. Moreover, the allowable sweeps are defined to be linear and to cover a half, a full and a one-and-a-half MDCT bin per time block, each in a downward and an upward direction version, resulting in 12 patterns to be stored additionally. For smallest possible memory requirements, sweep patterns can be stored only in one direction; the opposite direction can be derived by temporal mirroring of the pattern. According to Equation 8 (setting $m \in \{1, 3, 5...\}$), patterns involving half-bin sweep distances require post-processing rotations by $\varphi_0 + n\pi/4$.

If arbitrary patterns are chained in a temporal sequence, the start phase for the actual pattern at point t_0 of Figure 1 has to be adjusted (using the aforementioned rotations) and the predefined stop phase at point t_1 has to be stored for seamless continuation with the subsequent pattern. Sweeps that encompass half-bin sweep distances introduce the need for post-processing rotations by $\varphi_0 + n\pi/4$, for both sweep patterns and stationary patterns, since sweeps and stationary parts might be arbitrarily chained in a time sequence. A choice of $\varphi_0 = n\pi/2, n \in \mathbb{N}$ results in an easy-to-compute rotation performed averaging real and imaginary part of the pattern and a subsequent scaling by $\sqrt{2}$. Alternatively, all patterns can be additionally stored in a $\pi/4$ pre-rotated version and applied together with a trivial post-processing rotation by $n\pi/2, n = 1, 2, 3$ (see Table 1).

3.4. Example

In Figure 2 (a)-(f), the entire process, as described in Subsection 3.1, from pattern measurement up to pattern adaptation and patching is sketched. Firstly, a pattern is constructed by generating a sine or a sweep according to Equations 4 and 5. Then, the generated signal is transformed to ODFT frequency domain (a) to obtain a complex spectrum (b). Next, the complex pattern is trucated to its intended length (c) and stored in a table. Whenever the pattern is needed in order to synthesize a tonal signal portion, it is adapted to its target phase, and additionally it is compensated for the phase rotation induced by the spectral shift (d) as described in Subsection 3.3. Further, the time shift that is present in the CMDCT with respect to the ODFT is implemented by applying a post-twiddle according to Equation 3. Applying the post-twiddle can be done efficiently after summing up the contribution of all patterns to be patched into the spectrum (e). Lastly, the actual patching happens in the MDCT domain using only the real part of the adapted pattern. An IMDCT yields the desired time domain signal, the spectrum of which is depicted in panel (f).

4. RESULTS

4.1. Implementation

Figure 3 demonstrates a selection of different tone patterns for a typical low bit rate transform codec scenario using a 512 band MDCT, with a sine window, at a sampling frequency of 12.8 kHz, and a half-bin resolution for the tone patterns. From the top to the bottom panel, several normalized spectral ODFT tone patterns are plotted: sine on-bin, sine between-bin, sweep on-bin and sweep between-bin. All pattern types are stored in 4 variants: «on-bin» and «between-bin», «start

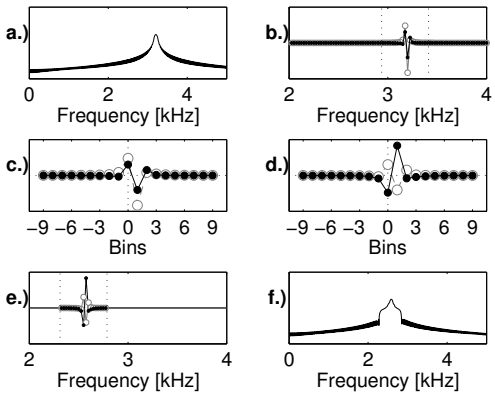

Fig. 2. Tone pattern patching process: pattern generation (a-b), pattern truncation (c), pattern adaption to target location and phase (d), and pattern patching (e-f). Real/imaginary part denoted by filled/empty dots.

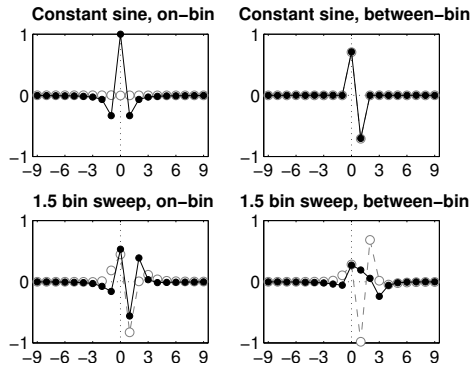

Fig. 3. Normalized spectral tone patterns: sine on-bin, sine between-bin, sweep on-bin, sweep between-bin (from top left to bottom right panel). Real/imaginary part denoted by filled/empty dots.

phase 0» and «start phase $\pi/4$» (pre-rotated patterns). Sweep patterns have additional 6 variants: «half», «full» and «one-and-a–half» bin sweep and «up» and «down» sweep direction. The total number of patterns to be stored is 4 times (1 stationary + 6 sweeps) and amounts to 28 complex patterns.

4.2. Quality and Computational Complexity

The signal quality that can be obtained by synthesizing truncated spectral patterns depends on the window type, which is usually already determined by the transform codec, and on the actual choice of pattern length, which can be adapted to the overall perceptual quality of the codec and the available resources (memory, computational complexity). Figure 4 shows the mean SNR as a function of pattern length for the sine window. In the scenario described in Subsection 4.1, truncating the patterns to e.g. 19 bins yields an average SNR of approximately 65 dB. If a lower SNR is acceptable, e.g. in a very low bit rate codec, already a pattern length of 5 bins might be sufficient.

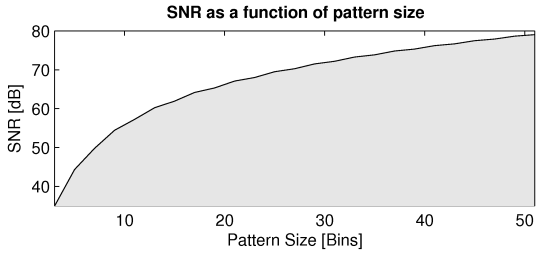

Fig. 4. Signal to noise ratio (SNR) of truncated tone pattern as a function of pattern length for sine window.

Additionally, the computational complexity of the proposed tone pattern synthesis was compared against the computational complexity of a bank of oscillators in time domain. It was assumed that a maximum of 20 sinusoidal tracks are active in a complete perceptual mono codec setup at a bit rate of 13.2 kbps. The computational workload was measured in the codec's fully instrumented C implementation in weighted millions operations per second (WMOPS) [9]. The items used for the measurements each contained at least one dominant tonal instrument with rich overtone content (e.g pitch pipe, violin, harpsichord, saxophon pop, brass ensemble). On average, the computational complexity of the tone pattern based synthesis is only 10% of the straightforward implementation using a bank of oscillators in the time domain.

5. CONCLUSIONS

We proposed a technique for the efficient parametric synthesis of sinusoids and sine sweeps in MDCT based audio coders through the application of spectral tone patterns that are adapted by post-processing phase rotations. For the actual synthesis of these tone patterns, the coder's IMDCT filter bank is co-used. Previous publications in the field relate to the generation of piecewise constant frequency tones based on DFT spectra [4], MDCT spectra in a perceptual codec environment [6] or in a bandwidth extension scenario [7]. However, the efficient generation of sweeps and their linkage to seamless tracks in MDCT domain has not been adressed yet, nor has the definition of sensible restrictions on the available degrees of freedom in the parameter space. Therefore, we detailed how the initial choice of the spectral resolution determines a lower cut-off frequency for perceptually appropriate tone generation, the storage memory demand and the computational complexity of the required pattern post-processing. We demonstrated that, by applying truncated tone patterns with an SNR of 65 dB in a low bit rate audio transform codec, computational complexity can be reduced down to 10% compared to the straight forward implementation of a bank of time domain oscillators.

6. REFERENCES

[1] N.H. van Schijndel and S. van de Par, "Rate-distortion optimized hybrid sound coding," in *Applications of Signal Processing to Audio and Acoustics, 2005. IEEE Workshop on*, oct. 2005, pp. 235 – 238.

[2] ISO/IEC 14496-3:2009, "Coding of Audio-Visual Objects, Part 3: Audio," Aug. 2009.

[3] Max Neuendorf, Markus Multrus, Nikolaus Rettelbach, Guillaume Fuchs, Julien Robilliard, Jeremie Lecomte, Stephan Wilde, Stefan Bayer, Sascha Disch, Christian Helmrich, Roch Lefebvre, Philippe Gournay, Bruno Bessette, Jimmy Lapierre, Kristofer Kjörling, Heiko Purnhagen, Lars Villemoes, Werner Oomen, Erik Schuijers, Kei Kikuiri, Toru Chinen, Takeshi Norimatsu, Chong Kok Seng, Eunmi Oh, Miyoung Kim, Schuyler Quackenbush, and Bernhard Grill, "MPEG Unified Speech and Audio Coding - The ISO/MPEG Standard for High-Efficiency Audio Coding of All Content Types," in *Audio Engineering Society Convention 132*, 4 2012.

[4] Nikolaus Meine and Heiko Purnhagen, "Fast sinusoid synthesis for MPEG-4 HILN parametric audio decoding," *Proc. of the 5 th Int. Conference on Digital Audio Effects (DAFx-02), Hamburg, Germany, September 26-28, 2002*, vol. 0, no. 0, 2002.

[5] Rade Kutil, "Optimized sinusoid synthesis via inverse truncated Fourier transform," *Trans. Audio, Speech and Lang. Proc.*, vol. 17, no. 2, pp. 221–230, Feb. 2009.

[6] Anibal J. S. Ferreira, "Perceptual coding using sinusoidal modeling in the MDCT domain," in *Audio Engineering Society Convention 112*, 4 2002.

[7] Anibal J. S. Ferreira and Deepen Sinha, "Accurate spectral replacement," in *Audio Engineering Society Convention 118*, 5 2005.

[8] Craig C. Wier, Walt Jesteadt, and David M. Green, "Frequency discrimination as a function of frequency and sensation level," *The Journal of the Acoustical Society of America*, vol. 61, no. 1, pp. 178–184, 1977.

[9] ITU-T Recommendation G.191, "Software tool library 2009 user's manual," 2009.